AMENDMENTS TO THE CLAIMS

Please cancel claims 11, 18, 32, 35-36, 41 and 53 without prejudice. Kindly amend claims 1, 12-14, 19, 25, 38, 40, 42, 44-45, 48 and 54 as shown in the following listing of claims. Kindly add new claims 55-101 as shown in the following listing of claims. The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (currently amended) An apparatus for correcting a prediction of a branch instruction in a pipeline microprocessor, the prediction predicting whether a branch condition specified by the branch instruction is satisfied in architected status flags of the microprocessor, the apparatus comprising:

   early status flags, corresponding to and storing a newer state than the architected status flags, said early status flags having a valid indicator; ~~and~~

   early branch correction logic, for correcting the prediction of the branch instruction if said valid indicator indicates said early status flags are valid and if said early status flags indicate the prediction is incorrect; and

   logic, coupled to said early status flags, for invalidating said early status flags if an early result of an instruction that precedes the branch instruction and that specifies modification of the status flags is invalid, wherein said logic copies the architected status flags to said early status flags and validates said early status flags if the microprocessor pipeline is flushed.

2. (original) The apparatus of claim 1, further comprising:

   late branch correction logic, coupled to said early branch correction logic, for correcting the prediction of the branch instruction if the architected status flags indicate the prediction is incorrect and if said early branch correction logic did not correct the incorrect prediction.

3. (original) The apparatus of claim 2, further comprising:

   an execution unit, coupled to said late branch correction logic, for generating values stored in the architected status flags.

4. (original) The apparatus of claim 3, wherein said values generated by said execution unit are generated based on valid source operands.

5. (original) The apparatus of claim 2, wherein said early branch correction logic is present in a first stage of the microprocessor pipeline and said late branch correction logic is present in a second stage of the microprocessor pipeline, wherein said second stage is subsequent to said first stage.

6. (original) The apparatus of claim 1, wherein the architected status flags store program-visible state of the microprocessor.

7. (original) The apparatus of claim 1, wherein the architected status flags are only updated based on results of instructions that the microprocessor has determined are guaranteed to complete.

8. (original) The apparatus of claim 1, wherein the architected status flags are always valid.

9. (original) The apparatus of claim 1, wherein said early status flags may be updated based on early results of instructions that the microprocessor has not yet determined are guaranteed to complete.

10. (original) The apparatus of claim 9, wherein an instruction is guaranteed to complete if the microprocessor determines that no instructions preceding said instruction are still capable of generating an exception and all branch instructions preceding said instruction are finally resolved.

11. (canceled)

12. (currently amended) The apparatus of ~~claim 11~~ claim 1, wherein said early result of said instruction that precedes the branch instruction is invalid if said early result is generated based on one or more invalid source operands.

13. (currently amended) The apparatus of ~~claim 11~~ claim 1, further comprising:

early execution logic, coupled to said logic, for generating said early result in response to said instruction that precedes the branch instruction.

14. (currently amended) The apparatus of claim 13, wherein said early execution logic ~~resides in a stage of the microprocessor pipeline~~ generates said early result prior to a ~~stage of the pipeline in which an~~ final execution unit ~~for executing~~ executes a final result of said instruction that precedes the branch instruction ~~resides~~.

15. (original) The apparatus of claim 13, wherein said early result of said instruction that precedes the branch instruction is invalid if said preceding instruction is not included in a set of instructions executable by said early execution logic.

16. (original) The apparatus of claim 15, wherein said set of instructions executable by the early execution logic is smaller than the set of instructions executable by the microprocessor.

17. (original) The apparatus of claim 13, wherein said early execution logic is further configured to generate memory addresses of instruction operands.

18. (canceled)

19. (currently amended) The apparatus of ~~claim 11~~claim 1, wherein said logic copies the architected status flags to said early status flags and validates said early status flags if no instructions in the pipeline between a first and second stage of the pipeline are instructions that require modification of the status flags.

20. (original) The apparatus of claim 19, wherein said first stage of the pipeline comprises a stage in which said early branch correction logic corrects the branch

instruction, wherein said second stage of the pipeline comprises a stage in which late branch correction logic corrects the branch instruction, wherein said late branch correction logic corrects the prediction of the branch instruction if the architected status flags indicate the prediction is incorrect and if said early branch correction logic did not correct the incorrect prediction.

21. (original) The apparatus of claim 1, wherein said early status flags are valid if said early status flags comprise architected status flag values correctly modified by each uncompleted instruction in the pipeline that is an early status flags-modifying instruction and that precedes the branch instruction.

22. (original) The apparatus of claim 1, wherein said correcting the prediction of the branch instruction comprises flushing the microprocessor pipeline above a stage in which the branch instruction resides and causing the microprocessor to fetch instructions at a corrected instruction address.

23. (original) The apparatus of claim 22, wherein said corrected instruction address comprises a next instruction sequentially following the branch instruction if the prediction predicted the branch instruction is taken.

24. (original) The apparatus of claim 22, wherein said corrected instruction address comprises a branch target address specified by the branch instruction if the prediction predicted the branch instruction is not taken.

25. (currently amended) The apparatus of claim 1, wherein the architected status flags comprise status flags stored in a register ~~substantially~~ conforming to an x86 architecture EFLAGS register.

26. (original) The apparatus of claim 1, wherein the branch instruction comprises an x86 architecture conditional branch instruction.

27. (original) The apparatus of claim 1, wherein the prediction of the branch instruction comprises a prediction of whether the branch instruction is taken or not taken.

28. (original) The apparatus of claim 1, wherein said early status flags indicate the prediction is incorrect if the prediction predicts the branch instruction is taken but the branch condition as indicated in said early status flags is false.

29. (original) The apparatus of claim 1, wherein said early status flags indicate the prediction is incorrect if the prediction predicts the branch instruction is not taken but the branch condition as indicated in said early status flags is true.

30. (original) The apparatus of claim 1, further comprising:

a first storage element, coupled to said early branch correction logic, for storing said early status flags; and

a second storage element, coupled to said first storage element, for storing said valid indicator.

31. (original) The apparatus of claim 1, wherein said early status flags are generated within an address generation stage of the pipeline microprocessor.

32. (canceled)

33. (original) The apparatus of claim 1, wherein the pipeline microprocessor is a scalar microprocessor.

34. (original) The apparatus of claim 1, wherein the pipeline microprocessor issues instructions in program order.

35-36. (canceled)

37. (original) The apparatus of claim 1, wherein the architected status flags store conditions specifiable by conditional program instructions of the microprocessor instruction set.

38. (currently amended) A pipelined microprocessor having a branch predictor for predicting whether a condition specified by a conditional branch instruction will be satisfied, comprising:

a storage element, for storing early status flags of the microprocessor for use in determining whether the branch condition is satisfied, wherein said early status flags correspond to and store a newer state than architected status flags of the microprocessor, wherein said early status flags are invalid if an instruction preceding the branch instruction specifies an operation that is not executable by early execution logic of the microprocessor; ~~and~~

branch correction logic, coupled to said storage element, for correcting the prediction if said early status flags are valid and said early status flags indicate the branch predictor mispredicted whether the condition specified by the conditional branch instruction is satisfied, and for generating an indication of whether said branch correction logic corrected the branch instruction for use by subsequent final branch correction logic; and

control logic, coupled to said storage element, for invalidating said early status flags if an early result of an instruction that precedes the branch instruction and that specifies modification of the early status flags is invalid, wherein said control logic copies said architected status flags to said early status flags and validates said early status flags if the microprocessor pipeline is flushed.

39. (original) The microprocessor of claim 38, wherein said early execution logic is in a same stage of the microprocessor pipeline as said storage element.

40. (currently amended) The microprocessor of claim 38, wherein said early status flags are invalid if modified based on results of an instruction preceding the branch instruction generated using invalid input operands.

41. (canceled)

42. (currently amended) The microprocessor of claim 38, further comprising:

an early register file, coupled to provide operands to said early execution logic for generation of a result of said preceding instruction, wherein said early status flags are modified in response to said result, wherein said early

status flags are invalid if said early register file provides an operand to said early execution logic that is invalid.

43. (original) The microprocessor of claim 42, wherein said early register file is further coupled to receive results generated by said early execution logic.

44. (currently amended) The microprocessor of claim 38, wherein if at least one input operand provided to said preceding instruction is invalid, said early status flags are invalid.

45. (currently amended) The microprocessor of claim 38, wherein said early status flags are generated within an address generation stage of the microprocessor pipeline.

46. (original) The microprocessor of claim 38, wherein the microprocessor is a scalar microprocessor.

47. (original) The microprocessor of claim 38, wherein the microprocessor issues instructions in program order.

48. (currently amended) A method for correcting a prediction of a conditional branch instruction early in a microprocessor pipeline, the method comprising:

determining whether early status flags are valid, wherein the early status flags correspond to and store a newer state than architected status flags of the microprocessor;

determining whether the prediction is incorrect based on whether the early status flags satisfy a branch condition specified by the conditional branch instruction; ~~and~~

correcting the prediction of the conditional branch instruction if the early status flags are valid and the prediction is incorrect based on whether the early status flags satisfy a branch condition specified by the conditional branch instruction;

invalidating the early status flags if an early result of an instruction that precedes the branch instruction and that specifies modification of the architected status flags is invalid; and

copying the architected status flags to the early status flags and validating the early status flags, in response to flushing of the microprocessor pipeline.

49. (original) The method of claim 48, further comprising:

generating the early status flags, prior to said determining whether early status flags are valid.

50. (original) The method of claim 49, wherein said generating the early status flags comprises:

generating the early status flags in response to generating an early result of an instruction preceding the conditional branch instruction.

51. (original) The method of claim 50, wherein the preceding instruction immediately precedes the conditional branch instruction by a single pipeline stage of the microprocessor.

52. (original) The method of claim 50, wherein said generating the early result comprises:

generating the early result in an address generation stage of the microprocessor.

53. (canceled)

54. (currently amended) A computer ~~data signal~~ program embodied ~~in~~ on a ~~transmission~~ computer-readable medium, comprising:

computer-readable program code for providing an apparatus for correcting a prediction of a branch instruction in a pipeline microprocessor, the prediction predicting whether a branch condition specified by the branch instruction is satisfied in architected status flags of the microprocessor, said program code comprising:

first program code for providing early status flags, corresponding to and storing a newer state than the architected status flags, said early status flags having a valid indicator; ~~and~~

second program code for providing early branch correction logic, for correcting the prediction of the branch instruction if said valid indicator indicates said early status flags are valid and if said early status flags indicate the prediction is incorrect; and

third program code for providing logic, coupled to said early status flags, for invalidating said early status flags if an early result of an instruction that precedes the branch instruction and that specifies modification of the status flags is invalid, wherein said logic copies the architected status flags to said early status flags and validates said early status flags if the microprocessor pipeline is flushed.

55. (new) An apparatus for correcting a prediction of a branch instruction in a pipeline microprocessor, the prediction predicting whether a branch condition specified by the branch instruction is satisfied in architected status flags of the microprocessor, the apparatus comprising:

early status flags, corresponding to and storing a newer state than the architected status flags, said early status flags having a valid indicator;

early branch correction logic, for correcting the prediction of the branch instruction if said valid indicator indicates said early status flags are valid and if said early status flags indicate the prediction is incorrect; and

logic, coupled to said early status flags, for invalidating said early status flags if an early result of an instruction that precedes the branch instruction and that specifies modification of the status flags is invalid, wherein said logic copies the architected status flags to said early status flags and validates

said early status flags if no instructions in the pipeline between a first and second stage of the pipeline are instructions that require modification of the status flags.

56. (new) The apparatus of claim 55, further comprising:

late branch correction logic, coupled to said early branch correction logic, for correcting the prediction of the branch instruction if the architected status flags indicate the prediction is incorrect and if said early branch correction logic did not correct the incorrect prediction.

57. (new) The apparatus of claim 56, further comprising:

an execution unit, coupled to said late branch correction logic, for generating values stored in the architected status flags.

58. (new) The apparatus of claim 57, wherein said values generated by said execution unit are generated based on valid source operands.

59. (new) The apparatus of claim 56, wherein said early branch correction logic is present in a first stage of the microprocessor pipeline and said late branch correction logic is present in a second stage of the microprocessor pipeline, wherein said second stage is subsequent to said first stage.

60. (new) The apparatus of claim 55, wherein the architected status flags store program-visible state of the microprocessor.

61. (new) The apparatus of claim 55, wherein the architected status flags are only updated based on results of instructions that the microprocessor has determined are guaranteed to complete.

62. (new) The apparatus of claim 55, wherein the architected status flags are always valid.

63. (new) The apparatus of claim 55, wherein said early status flags may be updated based on early results of instructions that the microprocessor has not yet determined are guaranteed to complete.

64. (new) The apparatus of claim 63, wherein an instruction is guaranteed to complete if the microprocessor determines that no instructions preceding said instruction are still capable of generating an exception and all branch instructions preceding said instruction are finally resolved.

65. (new) The apparatus of claim 55, wherein said early result of said instruction that precedes the branch instruction is invalid if said early result is generated based on one or more invalid source operands.

66. (new) The apparatus of claim 55, further comprising:

early execution logic, coupled to said logic, for generating said early result in response to said instruction that precedes the branch instruction.

67. (new) The apparatus of claim 66, wherein said early execution logic generates said early result prior to a final execution unit executes a final result of said instruction that precedes the branch instruction.

68. (new) The apparatus of claim 66, wherein said early result of said instruction that precedes the branch instruction is invalid if said preceding instruction is not included in a set of instructions executable by said early execution logic.

69. (new) The apparatus of claim 68, wherein said set of instructions executable by the early execution logic is smaller than the set of instructions executable by the microprocessor.

70. (new) The apparatus of claim 66, wherein said early execution logic is further configured to generate memory addresses of instruction operands.

71. (new) The apparatus of claim 55, wherein said logic copies the architected status flags to said early status flags and validates said early status flags if the microprocessor pipeline is flushed.

72. (new) The apparatus of claim 55, wherein said first stage of the pipeline comprises a stage in which said early branch correction logic corrects the branch instruction, wherein said second stage of the pipeline comprises a stage in which late branch correction logic corrects the branch instruction, wherein said late branch correction logic corrects the prediction of the branch instruction if the architected status flags indicate the prediction is incorrect and if said early branch correction logic did not correct the incorrect prediction.

73. (new) The apparatus of claim 55, wherein said early status flags are valid if said early status flags comprise architected status flag values correctly modified by each uncompleted instruction in the pipeline that is an early status flags-modifying instruction and that precedes the branch instruction.

74. (new) The apparatus of claim 55, wherein said correcting the prediction of the branch instruction comprises flushing the microprocessor pipeline above a stage in which the branch instruction resides and causing the microprocessor to fetch instructions at a corrected instruction address.

75. (new) The apparatus of claim 74, wherein said corrected instruction address comprises a next instruction sequentially following the branch instruction if the prediction predicted the branch instruction is taken.

76. (new) The apparatus of claim 74, wherein said corrected instruction address comprises a branch target address specified by the branch instruction if the prediction predicted the branch instruction is not taken.

77. (new) The apparatus of claim 55, wherein the architected status flags comprise status flags stored in a register conforming to an x86 architecture EFLAGS register.

78. (new) The apparatus of claim 55, wherein the branch instruction comprises an x86 architecture conditional branch instruction.

79. (new) The apparatus of claim 55, wherein the prediction of the branch instruction comprises a prediction of whether the branch instruction is taken or not taken.

80. (new) The apparatus of claim 55, wherein said early status flags indicate the prediction is incorrect if the prediction predicts the branch instruction is taken but the branch condition as indicated in said early status flags is false.

81. (new) The apparatus of claim 55, wherein said early status flags indicate the prediction is incorrect if the prediction predicts the branch instruction is not taken but the branch condition as indicated in said early status flags is true.

82. (new) The apparatus of claim 55, further comprising:

a first storage element, coupled to said early branch correction logic, for storing said early status flags; and

a second storage element, coupled to said first storage element, for storing said valid indicator.

83. (new) The apparatus of claim 55, wherein said early status flags are generated within an address generation stage of the pipeline microprocessor.

84. (new) The apparatus of claim 55, wherein the pipeline microprocessor is a scalar microprocessor.

85. (new) The apparatus of claim 55, wherein the pipeline microprocessor issues instructions in program order.

86. (new) The apparatus of claim 55, wherein the architected status flags store conditions specifiable by conditional program instructions of the microprocessor instruction set.

87. (new) A pipelined microprocessor having a branch predictor for predicting whether a condition specified by a conditional branch instruction will be satisfied, comprising:

a storage element, for storing early status flags of the microprocessor for use in determining whether the branch condition is satisfied, wherein said early status flags correspond to and store a newer state than architected status flags of the microprocessor, wherein said early status flags are invalid if an instruction preceding the branch instruction specifies an operation that is not executable by early execution logic of the microprocessor;

branch correction logic, coupled to said storage element, for correcting the prediction if said early status flags are valid and said early status flags indicate the branch predictor mispredicted whether the condition specified by the conditional branch instruction is satisfied, and for generating an indication of whether said branch correction logic corrected the branch instruction for use by subsequent final branch correction logic; and

control logic, coupled to said storage element, for invalidating said early status flags if an early result of an instruction that precedes the branch instruction and that specifies modification of the early status flags is invalid, wherein

said control logic copies the architected status flags to said early status flags and validates said early status flags if no instructions in the pipeline between a first and second stage of the pipeline are instructions that require modification of the status flags.

88. (new) The microprocessor of claim 87, wherein said early execution logic is in a same stage of the microprocessor pipeline as said storage element.

89. (new) The microprocessor of claim 87, wherein said early status flags are invalid if modified based on results of an instruction preceding the branch instruction generated using invalid input operands.

90. (new) The microprocessor of claim 87, further comprising:

an early register file, coupled to provide operands to said early execution logic for generation of a result of said preceding instruction, wherein said early status flags are modified in response to said result, wherein said early status flags are invalid if said early register file provides an operand to said early execution logic that is invalid.

91. (new) The microprocessor of claim 90, wherein said early register file is further coupled to receive results generated by said early execution logic.

92. (new) The microprocessor of claim 87, wherein if at least one input operand provided to said preceding instruction is invalid, said early status flags are invalid.

93. (new) The microprocessor of claim 87, wherein said early status flags are generated within an address generation stage of the microprocessor pipeline.

94. (new) The microprocessor of claim 87, wherein the microprocessor is a scalar microprocessor.

95. (new) The microprocessor of claim 87, wherein the microprocessor issues instructions in program order.

96. (new) A method for correcting a prediction of a conditional branch instruction early in a microprocessor pipeline, the method comprising:

determining whether early status flags are valid, wherein the early status flags correspond to and store a newer state than architected status flags of the microprocessor;

determining whether the prediction is incorrect based on whether the early status flags satisfy a branch condition specified by the conditional branch instruction;

correcting the prediction of the conditional branch instruction if the early status flags are valid and the prediction is incorrect based on whether the early status flags satisfy a branch condition specified by the conditional branch instruction;

invalidating the early status flags if an early result of an instruction that precedes the branch instruction and that specifies modification of the architected status flags is invalid; and

copying the architected status flags to the early status flags and validating the early status flags, in response to detecting a condition in which no instructions in the pipeline between a first and second stage of the pipeline are instructions that require modification of the status flags.

97. (new) The method of claim 96, further comprising:

generating the early status flags, prior to said determining whether early status flags are valid.

98. (new) The method of claim 97, wherein said generating the early status flags comprises:

generating the early status flags in response to generating an early result of an instruction preceding the conditional branch instruction.

99. (new) The method of claim 98, wherein the preceding instruction immediately precedes the conditional branch instruction by a single pipeline stage of the microprocessor.

100. (new) The method of claim 98, wherein said generating the early result comprises:

generating the early result in an address generation stage of the microprocessor.

101. (new) A computer program embodied on a computer-readable medium, comprising:

computer-readable program code for providing an apparatus for correcting a prediction of a branch instruction in a pipeline microprocessor, the prediction predicting whether a branch condition specified by the branch instruction is satisfied in architected status flags of the microprocessor, said program code comprising:

first program code for providing early status flags, corresponding to and storing a newer state than the architected status flags, said early status flags having a valid indicator;

second program code for providing early branch correction logic, for correcting the prediction of the branch instruction if said valid indicator indicates said early status flags are valid and if said early status flags indicate the prediction is incorrect; and

third program code for providing logic, coupled to said early status flags, for invalidating said early status flags if an early result of an instruction that precedes the branch instruction and that specifies modification of the status flags is invalid, wherein said logic copies the architected status flags to said early status flags and validates said early status flags if no instructions in the pipeline between a

first and second stage of the pipeline are instructions that require modification of the status flags.